

Trabajo práctico final

Circuitos Lógicos Programables

Especialización en Sistemas Embebidos

Fernando Nicolas Calvet

(fernando.n.calvet@gmail.com)

24/04/2024

versión A

Introduccion

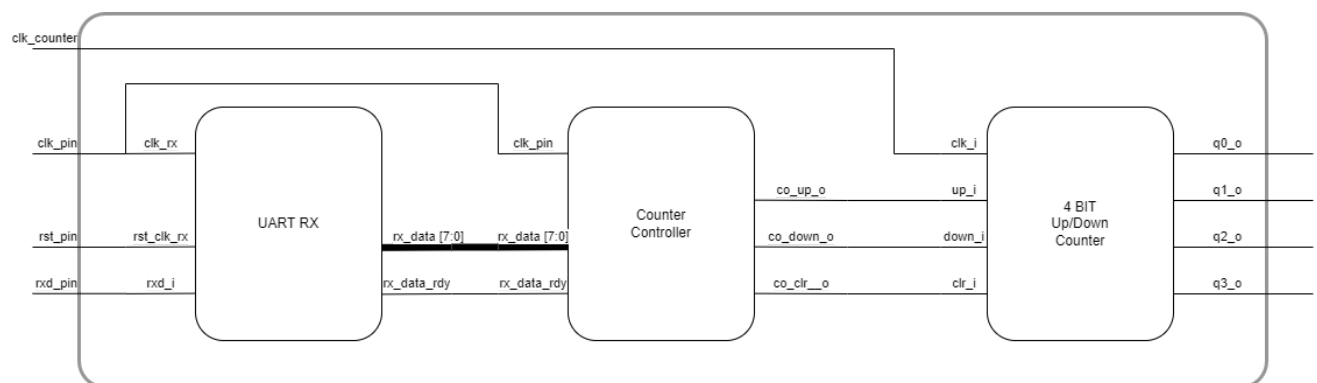
Este proyecto tiene como objetivo crear un sistema digital que actúe como un contador, capaz de incrementar, decrementar su valor en función de los comandos recibidos a través de la interfaz UART (Universal Asynchronous Receiver-Transmitter) y de un clock externo que será el responsable de incrementar o decrementar el contador segun la configuracion aportada por el controlador UART.

El diseño se implementa en lenguaje VHDL (Very High Speed Integrated Circuit Hardware Description Language), que es un lenguaje de descripción de hardware utilizado para modelar y simular circuitos digitales.

El proyecto implica varias etapas, incluida la simulación del diseño para verificar su correcto funcionamiento. Finalmente, se crea un documento que documenta todo el proceso, incluyendo una explicación detallada del diseño, diagramas de bloques, capturas de simulaciones y resultados de la implementación en la FPGA.

Este proyecto integra conceptos fundamentales de diseño de hardware digital, comunicación serial y tecnología FPGA para la creación de un sistema práctico y versátil. El contador desarrollado puede ser controlado y monitoreado a través de una interfaz serial como la UART, permitiendo su aplicación en diversas tareas de conteo y control en sistemas electrónicos.

Diagrama general del hardware a implementar



Descripción del funcionamiento

El módulo **UART RX** recibirá datos serie con la siguiente configuración:

- Baud Rate: 115200
- Bits: 8

- Bits de stop: 1
- Paridad: nula

En caso de ser un dato correcto con la configuración propia del dispositivo, entonces tomara como valido el dato y pasara el dato en forma de `std_logic_vector` al “**Counter controller**” junto con la señal de data ready.

El módulo `counter controller` examina el dato recibido y lo compara con una serie de palabras predefinidas que representan comandos válidos. Si el dato coincide con alguno de estos comandos, el `counter controller` cambia el estado de la señal de salida correspondiente. Los comandos válidos incluyen:

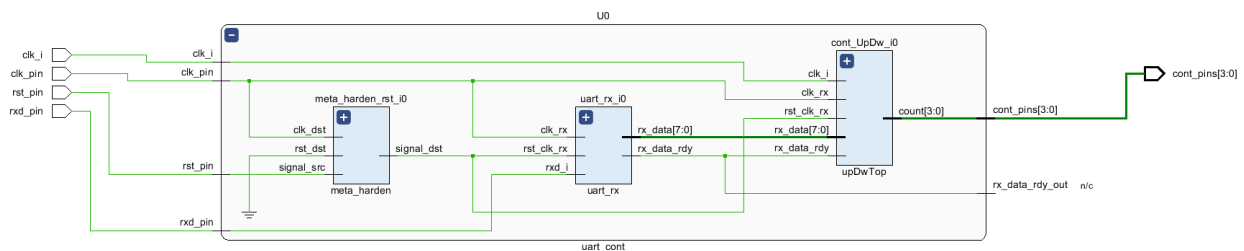
- "U" para incrementar (UP).
- "D" para decrementar (DOWN).
- "C" para restablecer (CLEAR).

Para el caso de UP y DOWN, tendrá prioridad UP en caso de que ambos estén en “1”, y si ambos 2 están en ‘0’, entonces el contador dejará de contar hasta en tanto vuelva a estar alguno de estos 2 valores en ‘1’.

El módulo `4 bit up/down counter` representa el contador final. Este módulo cuenta con un reloj independiente que controla su funcionamiento de conteo. Además de eso, recibe las señales de control generadas por el `counter controller` para determinar si debe incrementar, decrementar o restablecer su valor, según los comandos recibidos a través de la interfaz UART.

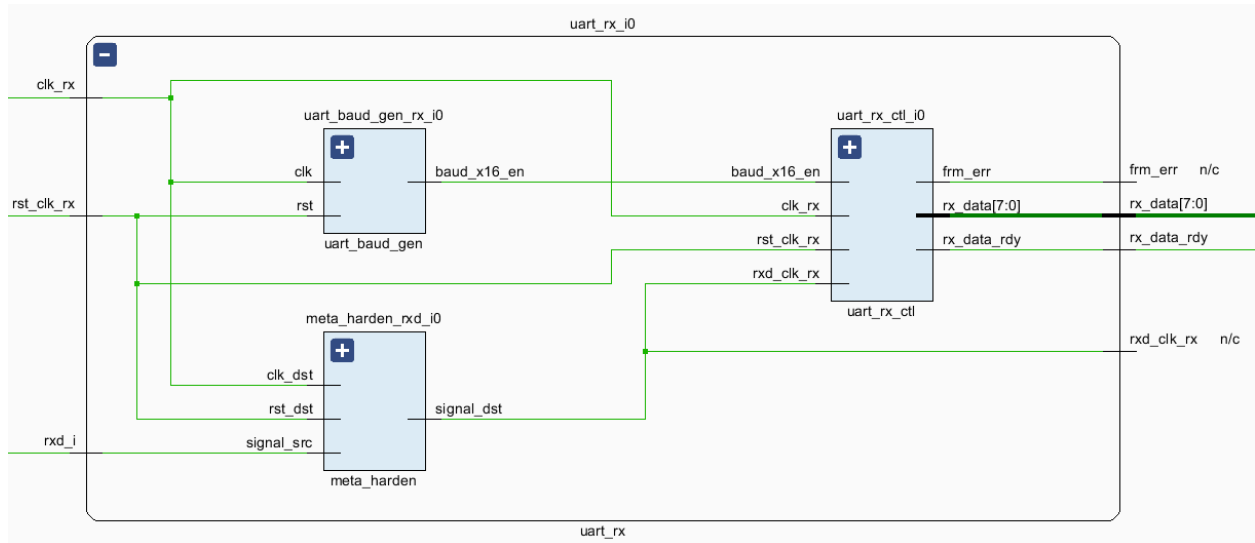
RTL analysis

Utilizando la herramienta disponible en Vivado, se llega al este resultado:

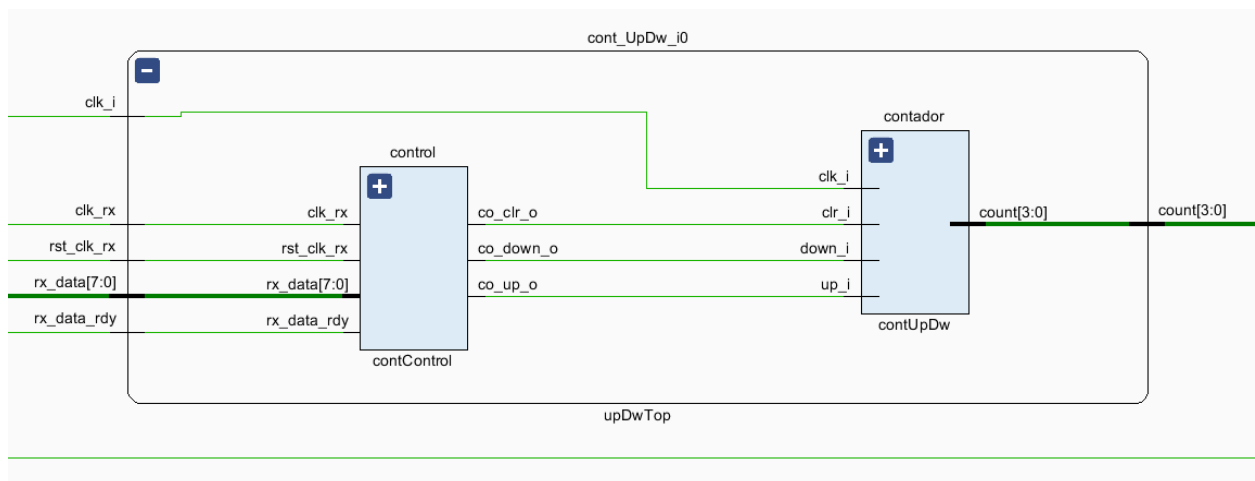


Los archivos de descripción de hardware `.vhd` se concibieron de forma tal que se pueda replicar el diagrama planteado anteriormente.

En donde por un lado se tiene lo referente al receptor UART (Se tomó los archivos utilizados en la práctica de la materia).



Y por otro lado se tiene el controlador y el contador propiamente dicho:



Simulacion

Para verificar el correcto funcionamiento del diseño completo, se desarrollaron "test benches" individuales para cada bloque y se realizó una simulación final del conjunto completo. A continuación, se presenta la simulación del sistema completo en lugar de los "test benches" individuales (enlaces a los "test benches" individuales se agregarán en breve).

Configuración de la Simulación

Clocks:

clk_pin: Clock general para todos los bloques (excepto el contador) con un período de 8 ns.

clk_i: Clock específico para el contador con un período de 20 μ s.

Entradas:

Rxd_pin: Entrada de datos serie.

Salidas:

Cont_pins: Pines de salida del contador.

Señales Internas:

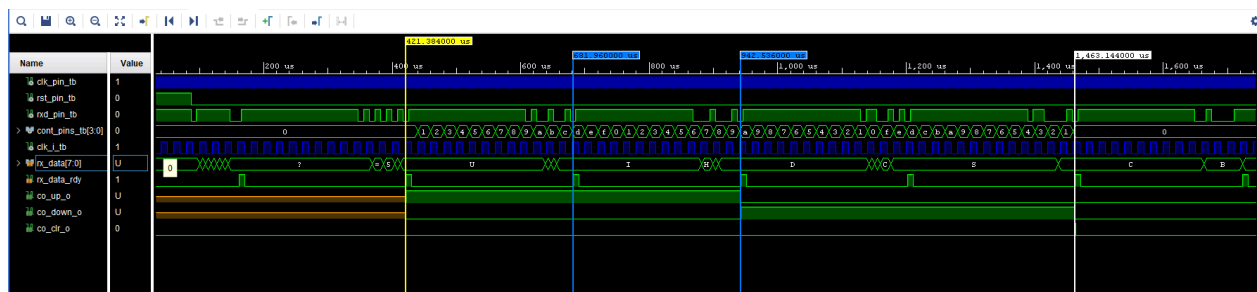
Co_up_o: Controla el incremento del contador.

Co_down_o: Controla el decremento del contador.

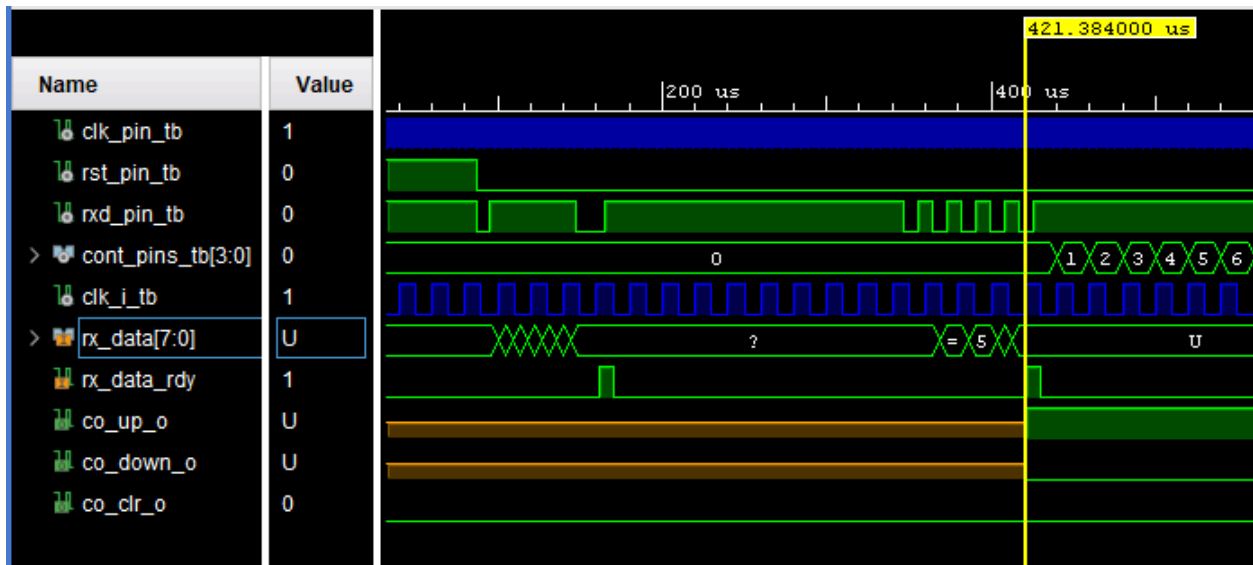
Co_clr_o: Resetea el contador a 0.

Resultados de la Simulación

La simulación del diseño completo demuestra el correcto funcionamiento de todos los bloques individuales y su interacción dentro de la estructura jerárquica. Los datos recibidos en Rxd_pin son procesados por el controlador, generando las señales Co_up_o, Co_down_o y Co_clr_o que controlan el contador. El contador actualiza su valor en función de estas señales y envía el resultado a los pines Cont_pins.

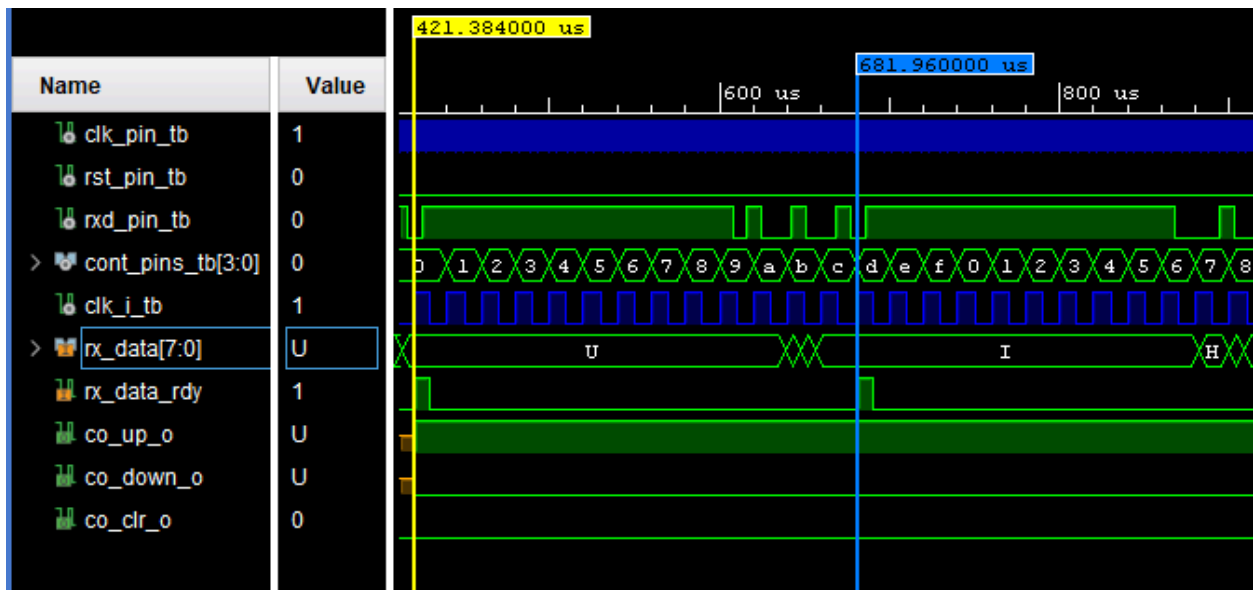


Marcador 1:



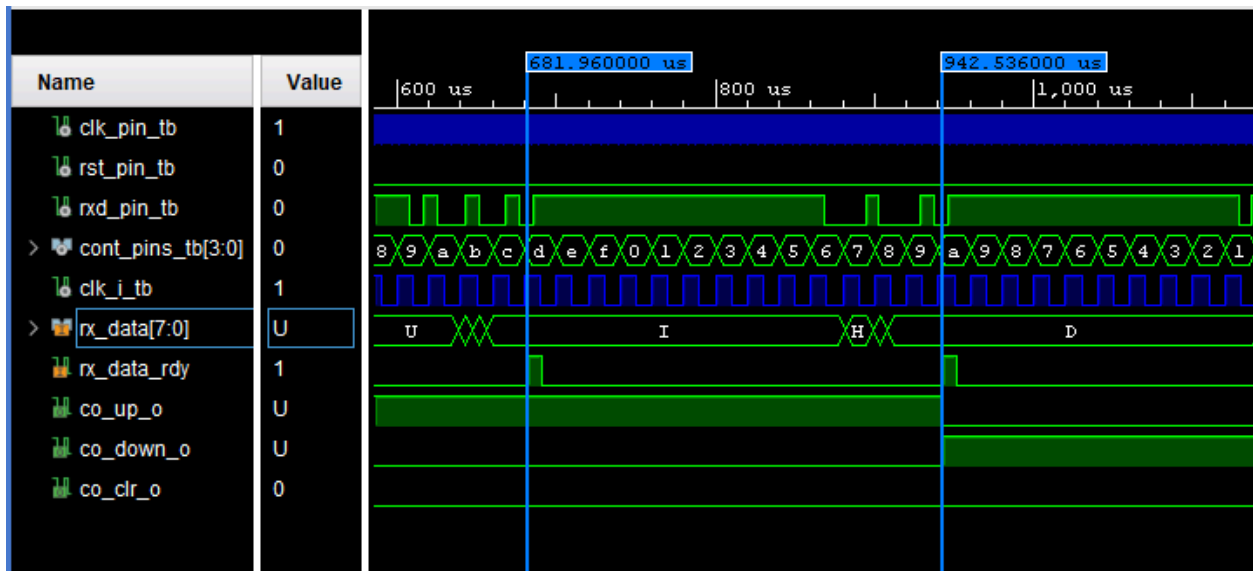
Se ven los valores iniciales asignados por el test bench. Se envia la letra 'U' por la puerta serie, se interpreta correctamente, se envia la señal de `rx_data_rdy`, el controlador configura la salida `co_up_o` en uno y el contador empieza a contar de forma ascendente a medida que llegan flancos positivos por `clk_i`.

Marcador 2:



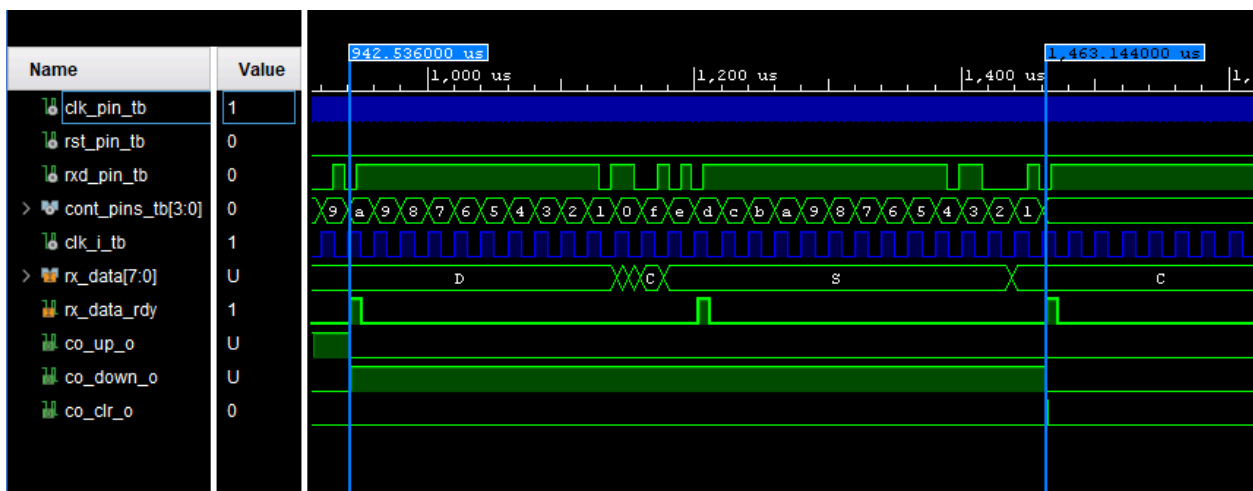
Se vuelve a enviar un dato por la puerta serie, en este cado la letra 'I'. Como no es válido, el controlador descarta este dato y no hay modificaciones en el comportamiento.

Marcador 3:



Se envía ahora la letra 'D' por la puerta serie. El controlador ahora cambia su configuración y pone en '1' la salida **co_down_o** y en '0' la salida **co_up_o**. El contador por su parte comienza a decrementar.

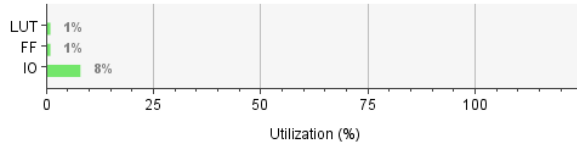
Marcador 4:



Por último se envía por serie la letra 'C' (clear). En este caso se envía un pulso por la salida **co_clr_o**. El contador se pone en '0' y deja de contar inmediatamente.

Tabla de uso de recursos

Resource	Utilization	Available	Utilization %
LUT	46	17600	0.26
FF	40	35200	0.11
IO	8	100	8.00



Name	Slice LUTs (17600)	Slice Registers (35200)	Bonded IOB (100)	BUFGCTRL (32)
uart_top	46	40	8	2
U0 (uart_cont)	46	40	0	0
cont_UpDw_i0 (up...	3	8	0	0
meta_harden_rst_i...	0	2	0	0
uart_rx_i0 (uart_rx)	43	30	0	0
meta_harden_rx...	0	2	0	0
uart_baud_gen...	8	8	0	0
uart_rx_cti_i0 (ua...	35	20	0	0

Conclusiones

Es importante destacar que el desarrollo de este proyecto ha requerido un buen entendimiento de los principios de funcionamiento del hardware digital y FPGA. Si bien es un proyecto sencillo, me ha servido para comprender el “lenguaje” y la forma de trabajo que requiere esta tecnología.

Posibilidades de mejora de implementación

Sería interesante realizar el contador de forma configurable tal que pueda utilizarse un “generic” para indicarle los bits de salida.

También podría mejorarse la implementación del testbench, ya que no se utiliza ninguna optimización al momento de enviar los datos simulados por la puerta serie.